Journal of Engineering Science and Technology
Vol. 15, No. 1 (2020) 032 - 045

EXPERIMENTAL INVESTIGATION OF INTEGRATED ID METHOD TO MITIGATE MESSAGE LOSS IN IOT CONTROL DEVICES

ANKIT KHARE*, RASHMI SHARMA, NEELU JYOTI AHUJA

School of Computer Science, University of Petroleum and Energy Studies (UPES), Bidholi Campus, Dehradun, Uttarakhand, India
*Corresponding Author: ankit.khare86@gmail.com

Abstract

Number of devices connected to the internet are rapidly growing and control devices are connected to the sensor to monitor certain conditions. An increase in the number of devices in the server causes high message loss for various messaging techniques. This requires effective method to minimize the message loss in the various devices. The aim of this research is to develop the model of integrate Identity (ID) technique to minimize message loss and test it in Internet of Things (IoT) environment. Integrated ID developed on the push message service of Message Queuing Telemetry Transport Protocol (MQTT). Integrated ID is the method of combining the hexadecimal number in the actual message and the ID is used to retrieve the data in the edge devices if the message is lost. The message consists the temperature data of the subject and order of ID is added in the message in hexadecimal value. The message ID in hexadecimal value is added in the prefix of the data, the device ID is represented in 2 bit, which is added after the temperature data. The control device checks the order of the message and found the missing data, then requested for the lost data. This condition requires very less memory to store the message order and supports a number of devices in the network. Experimental results show that proposed method transfer the first message within 0.86 s and have satisfactory latency.

Keywords: Internet of things, Message queuing telemetry transport protocol, Push protocol, Raspberry pi 3.

1. Introduction

Intern of things (IoT) integrates the physical world with a computer system that helps to operate the device smoothly even from the remote area [1]. The sensors are placed across the environment for monitoring purpose and the suitable action is taken after analysing the data. Presently, a number of devices equipped with sensors and processing devices, that helps in monitoring and controlling the various devices [2]. IoT is more useful in Healthcare application and involves in monitoring the patients remotely with considerable scalability, flexibility, and interoperability [3]. This device uses a cloud system to store and manage data that involves in the enhancement of resource utility of the devices [4]. In order to improve the function of the IoT, existing research and industries provide many frameworks, tools, and software. Some of the techniques have compatibility issues between the devices and complex implementation of comprehensive consumer applications [5]. A new communication method is needed to easily control different kind of smart devices by the same user.

The message passing between several devices requires one or more communication protocols between them. IoT devices having limited resources, so the protocols are designed to operate in low bandwidth, communication instability and high latency networks [6]. With the help of this protocol, the smart devices can smoothly communicate with other smart devices. The different communication protocols are MQTT, Constrained Application Protocol (CoAP), HyperText Transfer Protocol (HTTP), Representational State Transfer (REST) used to interface sublayer that allows the devices to engage and interact [7]. The communication endpoints are often special devices like sensor and actuators, which exchange the data to coordinate their operations. The communication between these smart devices defines the process of IoT [8], two most popular communication protocols are Message Queuing Telemetry Transport Protocol (MQTT) and Hypertext Transfer Protocol (HTTP) [9], [10]. MQTT protocol is one of the common technique used to transmit the message between the servers and to control the devices.

In this research, the Integrated ID technique is used to minimize the message loss in the server using message ID and device ID in the message. Raspberry Pi 3 is used to measure the temperature of the room or environment. MQTT is the standard protocol for transmitting the message to the server and devices, and this uses the push protocol method to send the message. The temperature is measured using the sensor device and the ID is added in the data that helps to get the message easily. The proposed method monitors the order of message passing in the network. If there any missing message is found, then the request message is sent to the respective device to resend the message.

The organization of the paper as follows, the section 2 explains literature survey, section 3 presents the proposed method and section 4 discussed about the experimental result.

2. Literature Review

Researchers have suggested several techniques to increase the efficiency of the messaging technique in IoT. The latest researches related to reliability of the message services in IoT system are analysed in this section.

Hwang et al. [11] designed and implemented the collaboration messaging system based on the MQTT protocol to maintain the message ordering. MQTT

protocol does not send the message in the order, and this creates the reliability problem in message transmission system. This technique transmits the message sequentially with little delay. This technique was tested on the simple message transition system and has to be evaluated in complex structures.

Agyemang et al. [12] proposed the device management system based on the resource of the system and service enablement architecture. This method managed the IoT environment by assigning the operation of the devices based on its resources to leverage Representational State Transfer (REST) architectural style. The performance and feasibility of this method were tested on the smart home application scenario. The technique was implemented using oneM2M protocol to gain the function through HTTP, APIs since the oneM2M protocol has been developed based on the resources and this process with resource constraint Device Management (DM) protocols. Therefore, this method performed based on the resource service in IoT application with simultaneous MQTT and LwM2M endpoint. The loss of data is not considered in this paper and transmission of the messages are needed to be more efficient. This technique has to be applied to some other method for the evaluation purpose.

Abdullah and Yang [13] presented a method for message scheduling with less energy consumption based on the recovery and backup node selection for IoT network. This message scheduling technique was simulated and evaluated in terms of node failure and replacement node. This method provided the energy efficient system and also reduced the node failure, but it requires more space to store the backup node and also need more computational time.

Roy et al. [14] constructed the gateway for load balancing solution for small talk static defined Wireless Sensor (WS) cluster to eliminate the short lifetime node participation and packet loss delay. The theoretical explanation is provided for the unusual behaviour of Wireless Sensor Network (WSN) gateways, formula to overcome the end-to-end delay and reduced the packet loss. The experimental environment was created to evaluate the performance with the help of private cloud platform Arduino Uno, Raspberry Pi 3, MQTT broker, Amazon web services, and sensor devices in their Mobile Cloud computing laboratory. The evaluation of the result showed that the message payload and packet transfer are high compared to the existing method. This technique helped to reduce the message loss and delay of the message transmission in the IoT environment but failed to retrieve the lost message from the receiver.

Madaan et al. [15] analysed the privacy threat of information linkage, technical and legal method to represent the heterogeneous IoT ecosystem. They discussed about the information linkage in the data integration. They developed a method to provide the awareness to stack holders and protect the subjects about the privacy threat. The method increases the security in the system and message loss is high.

Fremantle and Aziz [16], propose a model for IoT based on the OAuth2 protocol that protect and maintain the identity of the user and devices. The identity of the users was not shared with third party application and provided the model to connect to the third party application. The user's data are stored in the cloud, secured it with specific protocols, and evaluated with sample third party app. This method was compared with other related research and this showed that the proposed method has the considerable performance. This method involved in cloud storage for user identity and third party application, which increase the message transmission time.

Many researchers on IoT considered the back up the data to minimize message loss that requires more storage and increase the computation time. To solve above-mentioned problems, the integrated ID technique is proposed and evaluated in the IoT network.

3. Proposed Methodology

Internet linked devices growing over the years, now billions of devices, including smartphone, sensors, energy meter etc., are connected through the internet. This causes the data traffic and which leads to message loss in control devices. An efficient method is needed to transfer the data without loss and it has to adopt in the lightweight environment. The Integrated ID technique is proposed in this research to transfer the data effectively and flow diagram of Integrated ID is shown in Fig. 1. The room temperature is taken as a monitoring factor to test the effectiveness of the proposed method. Sensor devices measure the room temperature and transfer the data to the server and server sends the data to the control devices. The control devices turn on/off those sensor devices using message transferring system. The integrated ID technique helps to recover the lost message from the control device. The ID number of messages generated in the hexadecimal order and transmit along with the temperature data and device ID is attached to the last part of the data. If the data lost, the control devices send the request signal to the specific devices to recover the data. This method is clearly explained in following section.

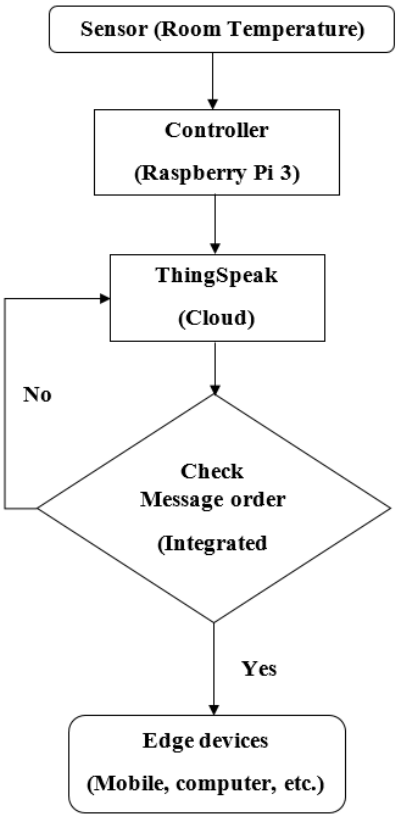

Fig. 1. Flow diagram of Integrated ID.

3.1. IoT Devices

The two types of IoT devices such as controller and sensor used in this experiment for message transmission through MQQT technique. The Raspberry Pi 3 is used as a controller and Arduino is used as the sensor module in the method. The proposed method is analysed in the transmission technique through MQQT protocol. Arduino collects the data of room temperature and Raspberry Pi 3 is used to show the data or control the sensor devices like turning on/off. Mostly smartphones are used as both control and sensor devices, it can be connected to more control devices for large applications. To maintain the reliable message transmission between sensor and control devices, the ratio of sensor device and control device is set with 1:N devices.

3.1.1. Arduino

Arduino is the open source board consists of the physical programmable memory (microcontroller) and the code can be uploaded to the Arduino by mean of software (IDE). Arduino does not need the separate hardware to upload the code. This follows most standard format to break the function of microcontroller, helps to access more package. This is popular open source electronic board and this is suitable to evaluate the proposed method [17].

3.1.2. Raspberry pi 3

Raspberry Pi 3 is the credit card size computer, which supports many functions from word or spreadsheet to games. This device supports keyboard, mouse, display and execute it on the Linux distribution. Raspberry Pi 3 model B is the latest version, which is low cost and this is used to test the proposed method [18].

3.2. Message queue telemetry transport (MQTT)

International Business Machines (IBM) Corporation developed MQTT protocol, which is a both instant messaging and lightweight broker messaging protocol [19]. This technique runs on any platform and supports most popular programming language. This messaging protocol is based on the pushing message technique and many enterprises used this protocol in an Android phone and this pushing protocol is in server-side. It is suitable for the mobile pushing message due to its simplicity and scalability. MQTT is widely used in the mobile pushing message [20] and its technique overview is shown in Fig. 2.

MQTT protocol works at the application level, especially designed for resource constrained devices [14, 21]. The publishing and subscribing technique is followed in this protocol, as a broker device. For instance, if the client sends the message denoted as M related to the topic T, then the message is sent to the all client, subscribed to the topic T. Similar to the HTTP, MQTT process is based on the Transmission Control Protocol (TCP) with Internet Protocol (IP) as its underlying layers. The reliability of the protocols is ensured by the three Quality of Service (QoS) levels. At the level 0, the message was delivered at once without acknowledging and in the level 1, acknowledge was received. The four-way handshake mechanism executes to deliver the message at level 2.

MQTT expose as the REST resource and it will not retain the states, the broker defines the state. The last payload is identified as the resource value and after the

analysis; the topic can be exposed by changing the single value. The topic is created based on the pure REST style with the help of HTTP POST and request for topic contains both topic name and payload. For instance, the topic's name is taken as temperature, and then the process is conducted as follows:

- Analysis the last published value by executing an HTTP GET request at /topics/temperature;
- Publish a value on the topic by executing an HTTP PUT
- The HTTP protocols allow the proper querying of the state of MQTT topic, this can be helped by using the HTTP caching protocol, contains the Last-Modified headers and ETag, clients can safely poll the broker about the state of the topic. The method provides only the best level of Qos because this use true nature of the HTTP protocol.

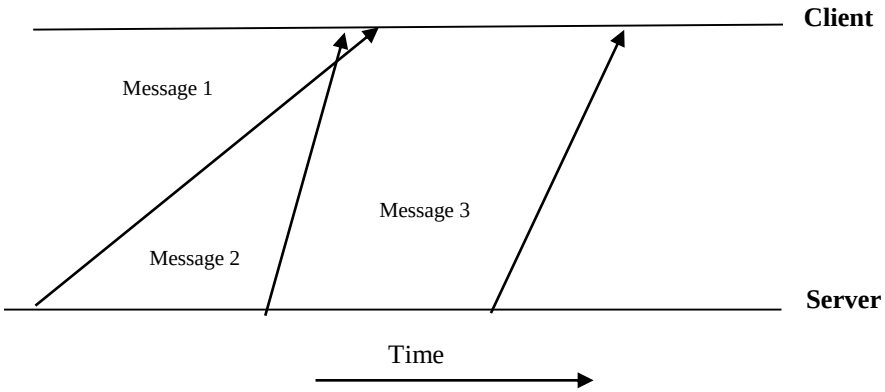

Fig. 2. Overview of MQTT method.

3.3. Integrate ID method

The sensor devices send the data to the cloud and cloud passes the received message to the control devices. The control devices turn on/off the sensor devices and this is used in various fields like medical, home usage, etc. The sensor sends the message to the server and there might be a loss of data in the server. Therefore, the message is added with hexadecimal numbers for analysis the order of the message and based on that missing data is requested. ID denotes the order of the message and in the last part of the message from the sensor devices is added with 2-bit data to address the devices. There might be many devices connected to the control devices. The device number has been used to identify missing data and this helps to understand easily the missing data from the device. The requested signal then sent from the cloud to that device to get the data. The ID number is generated for each message before publishing the data with the device number. The data field number is in the manner of hexadecimal numbers and the value is generated from 0000. Therefore, this has the capacity to transfer many data.

MQTT is the topic-based protocol, which uses the character strings to support the hierarchical topics. This technique can facilitate the multiple topics and it discussed with examples. For instance, temperature monitoring in floor "F4" of the building in the Room "R52" and its data transferred in the hierarchical form

"wsn/sensor/F4/R52/temperature". The data separated using "/" and wsn denotes the Wireless Sensor networks. The data are replaced by using a wildcard as "wsn/sensor/F4/+/temperature", which can now access any sensor to monitor the different room temperature in the 4th floor.

This technique includes the data, such as "wsn/sensor/F2/R248/xxxxtemp eraturexx", as first four digits represent the message and last 2-bit denotes the device. If many devices connected to the control devices, then the message from the devices are identified using its device ID. It helps to recover the data by knowing the missing data from the device using message order.

The IoT sensor devices data is attached the hexadecimal value to the temperature data in prefix and 2-bit is added as the device ID in the last, which contains the order of messages. This helps to understand the message order and request made for the missing data. These message order, then attached to every message send to the server and message can be easily noted. The Raspberry Pi request the message with the device ID helps to understand the respective device. If the message has been missed in the devices, then it requests to get the messages. The program is stored in the control devices and the server helps to analysis the data order. The server checks the number and it finds the missing data. Then it requests the sensor to send the message. This process is clearly explained by the Figs. 3 and 4.

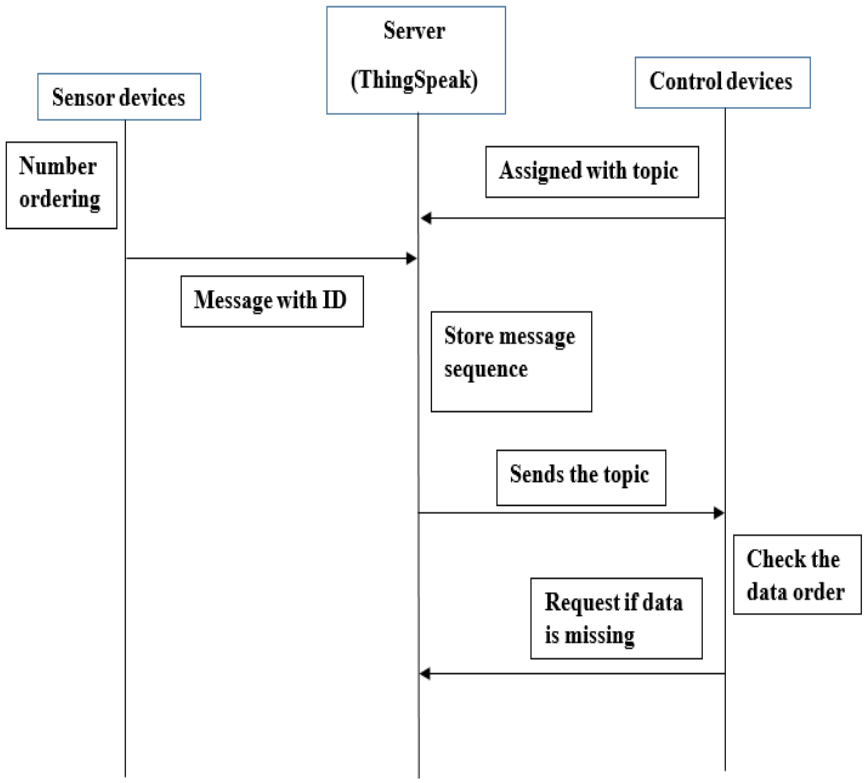

Fig. 3. Message transaction between sensor device to control device.

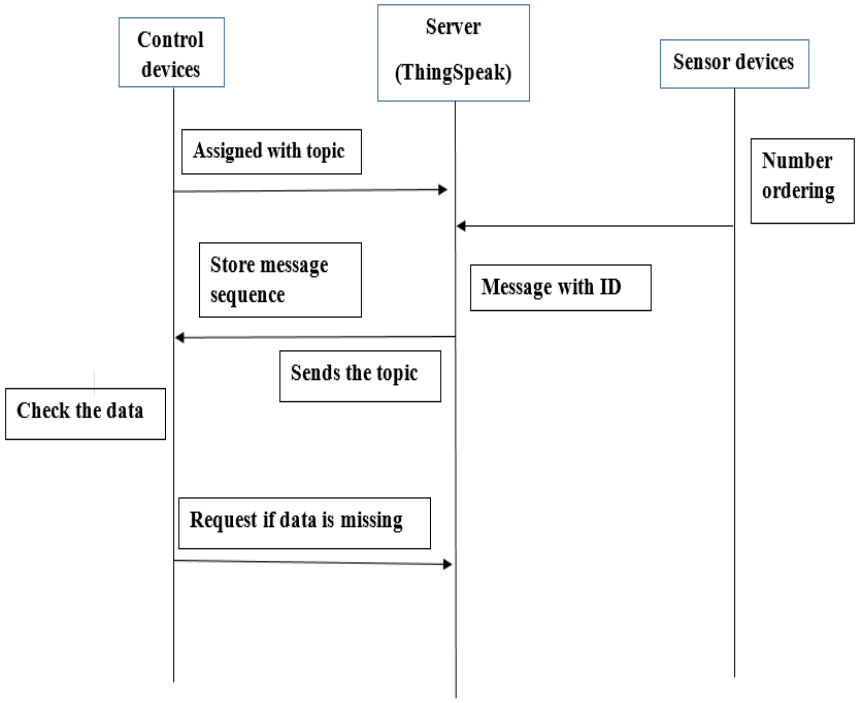

Fig. 4. Communication between control device and sensor device.

3.4. Mathematical derivation of integrated ID method

Once the temperature has been measured by the sensor in the room of 254 in the 4th floor of the building, the data has been combined with the ID in the control devices. This function is represented in the Eq. (1).

$$I_M . M . I_D \rightarrow I_M M I_D \tag{1}$$

From the Eq. (1), M denotes the message and I_M denote the identity of the message, I_D denotes the identity of the devices that is used to check the order of the message.

For example, if the message is sent as $24°C$ and this the first message send by the sensor then the message is added with ID, as in Eq. (2).

$$(0001).(24).(01) \rightarrow 00012401 \tag{2}$$

Equation (2) can be generally denoted as in Eq. (3). The first x denotes the digits of the message identity in the data and the y denotes the digits of the device identity.

$$xxxx.tt.yy \rightarrow xxxxttyy \tag{3}$$

The digits of the x and y can be increase depends on the monitoring time and number of devices in the room. If the temperature monitoring is plotted in the green field, there are more of sensor devices is need to plotted in the environment and their ID digits can be increased. The $xxxx$ is generated in the hexadecimal and yy is generated in the binary sequential order.

The control devices add the identity in the message, as this message can be send to the server. The message is send to the ThingSpeak and this will check the

message transmission in the system. In the ThingSpeak server, the topic T has been checked and based on the topic the data is send to the edge devices. The topic T example is shown in the Eq. (4).

$$wsn/sensor/F4/R52/data \quad (4)$$

where wsn denotes the network type, $sensor$ is the temperature sensor, $F4$ denotes the 4th floor, $R52$ denotes the 52nd room in the building and $data$ is given in the Eq. 3. The data is send to the edge devices; the message order is check in the edge devices. If there are any order missing in the message, then the request has been send to the server to resend the message. This method helps to reduce the message loss in the system.

4. Experimental Result

Most of the home appliance is connected through the Internet and controlled remotely using edge devices. The number of devices connected to the internet is growing high and it needs reliable messaging technique. The aim of the method is to reduce the loss of message by analysing the order of messages. The proposed method is evaluated using the Raspberry pi 3 and the message is sent through the devices. The unstable conditions created in the clumsy network simulator [11] and the message is analysed on the both sides of the devices. Table 1 shows the devices and its supported applications.

Table 1. Environment setup.

Item	Server	Sensor devices	Control devices
Operating System (OS)	-	Ubuntu 16.04 on Raspberry Pi 3	Microsoft Windows 10
Relational Database Management System (RDBMS)	Thingspeak	-	-
Message broker	Mosquitto v1.4.9	Mosquitto v1.4.9	Mosquitto v1.4.9

The latency of the proposed method is calculated for different messages that is presented in Fig. 5. Latency is lower for the first messages and it varies with different messages. This latency is considerable for the IoT application and this measured in the milliseconds. In the research [16], OAuthing transmitted the message in a secure manner and this involved in protecting the data. As these methods do not share the data with third-party application and additionally provide privacy to the user. This technique is compared with OAuthing research [16], in terms of time, latency, and program memory. Fremantle and Aziz [16] involved in user registration and then tried to connect the client and the user. The message was assigned with the ID and continuously monitored. The message loss minimized with the help of the token and the data stored in the separate cloud.

The connection time is based on the device initialization, and the evaluated connection time of different method is compared with other techniques in research [16]. Mosquitto technique has the connection time of 24.5ms and the integrated ID has the connection time of 22.95s. This time is less compared to existing methods.

The existing method of OAuthing [16] involved in initializing the third party app and other gateways, caused a rise in the connection time. The proposed method of integrated ID will not allow third party apps to involve, so processing time decreased. The IDs were stored in the variable memory of the device instead of cloud storage, which results in decreasing the latency of the proposed method. Figure 6 shows connection time of various techniques. The first connection time is much higher for the OAuthing method due to its privacy protection algorithm. The further connection of OAuthing having less computational time of 35.9ms.

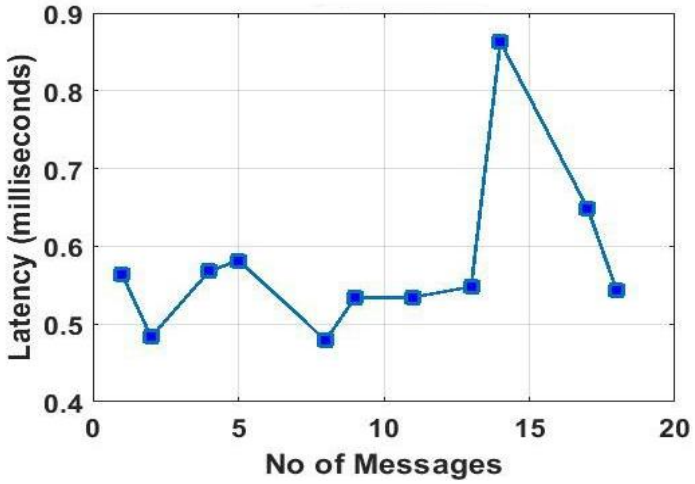

Fig. 5. Latency for number of messages.

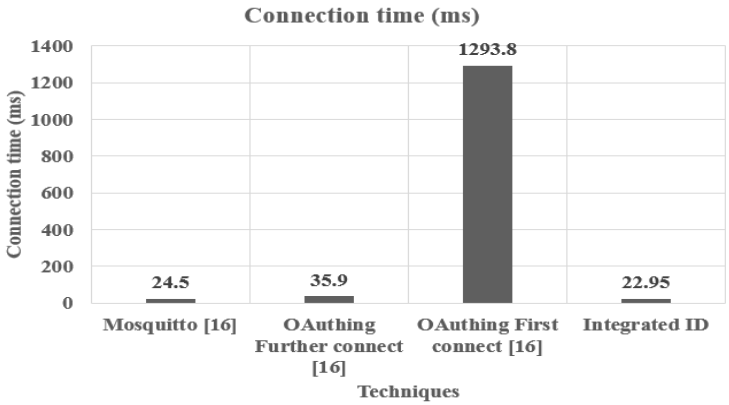

Fig. 6. Connection time for different methods.

The program memory gives the space required by the method to store and less program memory helps to provide more space for variable memory. The program memory of the proposed method and state of art method is shown in Fig. 7. The Integrated ID method requires lower memory than the existing method OAuThing [16]. The existing method store the data in the cloud and the integration of the cloud increases the program memory. The Integrated ID technique uses the numerical data for identifying message loss and this requires very less space that can be stored

in the variable memory of the devices. The proposed method stores and process the numerical data (device ID) for identification which requires less time to process than other existing methodologies with other information (user name, token, etc.).

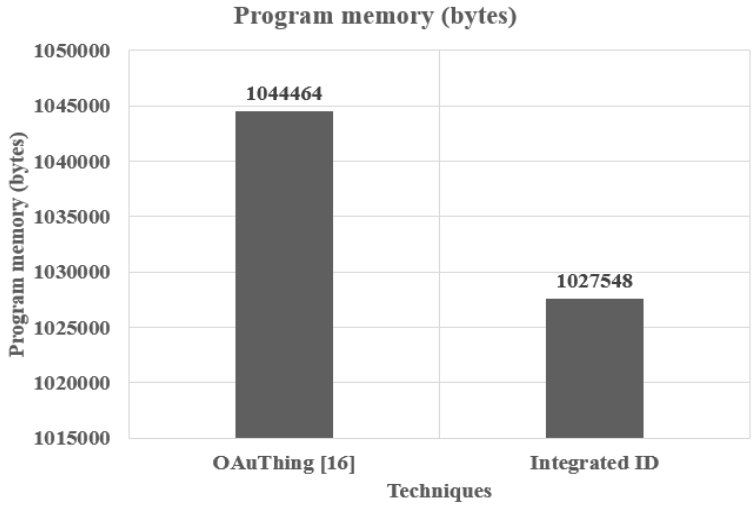

Fig. 7. Program memory size for proposed method.

IoT messaging technique is needed to transfer the messages across a number of devices and some devices may have lower resource. These kind of resources needs the lightweight program for its function. This Integrated ID method shows that the proposed method has low program memory compared to the other method.

The time requires for first message is calculated and compared with the state-of-art method. The comparison of time taken to process the first message between existing and proposed is shown in Table 2. The proposed method requires less time compared to other methods and the proposed method requires only 0.86 seconds for transmission of first messages. The existing methods involve in registering the data before transferring the message. The device ID in the Integrated ID identifies the user, as it helps to improve the performance of the message process in less time. This helps to reduce the time taken for the first message to process and achieves low time for first message compare to existing methods.

The wrong message rate is measured for the Integrated ID and the common MQTT technique in the Table 3. This shows that the proposed method has the zero wrong message rate and it also has high efficiency. However, the proposed integrated ID has been tested in the unstable network and this shows that the wrong message rate in 9%.

Table. 2 Time take by first message to process

Technique	Time (s)
MQQT [16]	12.03
OAuThing [16]	12.71
Integrated ID	0.86

Table 3. Wrong message rate of the Integrated ID.

Methods	Wrong message rate (%)
MQTT	4
Integrated ID	0

This shows that the proposed method has higher performance compared to the state-of-art method. This method can be applied to the IoT technique for efficient transfer of message, helps in monitoring the environment.

5. Conclusions

This research aims to minimize the loss of message in the IoT environment without affecting the performance of the system. IoT technology used in many fields, which involves in lot of devices, connected to the control devices. The integrated ID is proposed in this method and this method attaches the ID number with the message. As it helps to find the lost message during transmission in the control device and request the server to send the message again. The experimental result showed that this technique reduced the message loss with less execution time. Many IoT devices are a resource constraint, so there is a need of lightweight protocol. The program memory of the integrated ID is less compared to the conventional method and has much flexible for many devices. This method helps to monitor more data with less message loss. This helps to retrieve the message from the server and uses the less storage size in the devices. The execution time of the integrated ID technique achieved 0.86s with less storage memory. The contribution of the method is as follows

- The proposed method has the lower wrong message rate than the traditional message exchange method.
- The connection time of the method is low and efficient in the message exchange method.
- The message retrieval is very convenient and can be applied to the large field for the monitoring the environment.
- As the integrated ID method is light weighted and this is suitable for the majority of devices in the IoT.

In the future work, the security method can be developed in message transmission that helps to protect the privacy of the user.

References

1. Antunes, M.; Gomes, D.; and Aguiar, R.L. (2018). Towards IoT data classification through semantic features. *Future Generation Computer Systems*, 86, 792-798.
2. Paulraj, G.J.L.; Francis, S.A.J.; Peter, J.D.; and Jebadurai, I.J. (2018). Resource-aware virtual machine migration in IoT cloud. *Future Generation Computer Systems*, 85, 173-183.
3. Woo, M.W.; Lee, J.; and Park, K. (2018). A reliable IoT system for personal healthcare devices. *Future Generation Computer Systems*, 78, 626-640.

4. Nastic, S.; Truong, H.L.; and Dustdar, S. (2015). Sdg-pro: a programming framework for software-defined iot cloud gateways. *Journal of Internet Services and Applications*, 6(1), 21.
5. Batalla, J.M.; and Gonciarz, F. (2018). Deployment of smart home management system at the edge: mechanisms and protocols. *Neural Computing and Applications*,31(5), 1-15.
6. Rinne, J.; Keskinen, J.; Berger, P.R.; Lupo, D.; and Valkama, M. (2017). Viability bounds of M2M communication using energy-harvesting and passive wake-up radio. *IEEE Access*, 5, 27868-27878.
7. Khaled, A.E.; and Helal, S. (2018). Interoperable communication framework for bridging RESTful and topic-based communication in IoT. *Future Generation Computer Systems*, 92, 628-643.
8. Abreu, D.P.; Velasquez, K.; Curado, M.; and Monteiro, E. (2017). A resilient Internet of Things architecture for smart cities. *Annals of Telecommunications*, 72(1-2), 19-30.
9. Al-Ali, A.R.; Zualkernan, I.A.; Rashid, R.; Gupta, M.; and Alikarar, M. (2017). A smart home energy management system using IOT and big data analytics approach. *IEEE Transactions on Consumer Electronics*, 63(4), 426-434.
10. Said, O.; Albagory, Y.; Nofal, M.; and Raddady, F.A. (2017). IoT-RTP and IoT-RTCP: adaptive protocols for multimedia transmission over internet of things environments. *IEEE Access*, 5 , 16757-16773.
11. Hwang, H.C.; Park, J.; and Shon, J.G. (2015). Design and implementation of a collaboration messenger system based on MQTT protocol. *Proceeding of International Conference on Advances in Computer Science and Ubiquitous Computing*, Singapore, 513-519.
12. Agyemang, B.; Xu, Y.; Sulemana, N.; and Hu, H. (2018). Resource-oriented architecture toward efficient device management for the Internet of Things. *Journal of Ambient Intelligence and Humanized Computing*, 1-13.
13. Abdullah, S.; and Yang, K. (2014). An energy efficient message scheduling algorithm considering node failure in IoT environment. *Wireless personal communications*, 79(3), 1815-1835.
14. Roy, D.G.; Mahato, B.; De, D.; and Buyya, R. (2018). Application-aware end-to-end delay and message loss estimation in Internet of Things (IoT)-MQTT-SN protocols. *Future Generation Computer Systems*, 89, 300-316.
15. Madaan, N.; Ahad, M.A.; and Sastry, S.M. (2018). Data integration in IoT ecosystem: information linkage as a privacy threat. *Computer Law & Security Review*, 34(1), 125-133.
16. Fremantle, P.; and Aziz, B. (2018). Cloud-based federated identity for the Internet of Things. *Annals of Telecommunications*, 73(7-8), 415-427.
17. Ali, A.S.; Zanzinger, Z.; Debose, D.; and Stephens, B. (2016). Open Source Building Science Sensors (OSBSS): A low-cost Arduino-based platform for long-term indoor environmental data collection. *Building and Environment*, 100, 114-126.
18. Foster, S.W.; Alirangues, M.J.; Naese, J.A.; Constans, E.; and Grinias, J.P. (2019). A low-cost, open-source digital stripchart recorder for chromatographic detectors using a Raspberry Pi. *Journal of Chromatography A*, 1603, 396-400.

19. La Marra, A.; Martinelli, F.; Mori, P.; Rizos, A.; and Saracino, A. (2017). Improving MQTT by inclusion of usage control. *Proceeding of International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*, China, 545-560.
20. Luzuriaga, J.E.; Perez, M.; Boronat, P.; Cano, J.C.; Calafate, C.; and Manzoni, P. (2016). Improving MQTT data delivery in mobile scenarios: Results from a realistic testbed. *Mobile Information Systems*, 2016.
21. Santamaria, A.F.; De Rango, F.; Serianni, A.; and Raimondo, P. (2018). A real IoT device deployment for e-Health applications under lightweight communication protocols, activity classifier and edge data filtering. *Computer Communications*, 128, 60-73.